****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[Date]** | **[Your Name]** |  |

## Client



## Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Libby Felkay

## Algorithm Cipher

Provide a brief, high-level overview of the encryption algorithm cipher.

A cipher is a method or algorithm used to encrypt or decrypt data. Encryption turns easily readable data such as plaintext to ciphertext which needs to be decoded to be understood. Ciphers are used for data security and confidentiality. If you handle sensitive data, and someone has gained unauthorized access, encryption makes it much harder to steal said data.

Explain the purpose of the cipher's hash functions and bit levels?

Hash functions are used to ensure data integrity by generating a unique hash value for data. They are not encryption algorithms by themselves, but are often used with encryption to verify data integrity.

The bit level (256-bit in this instance) indicates the length of the encryption key. A higher bit level increases security, but comes at the cost of performance. SHA-256 is chosen here for its balance of security and performance.

Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.

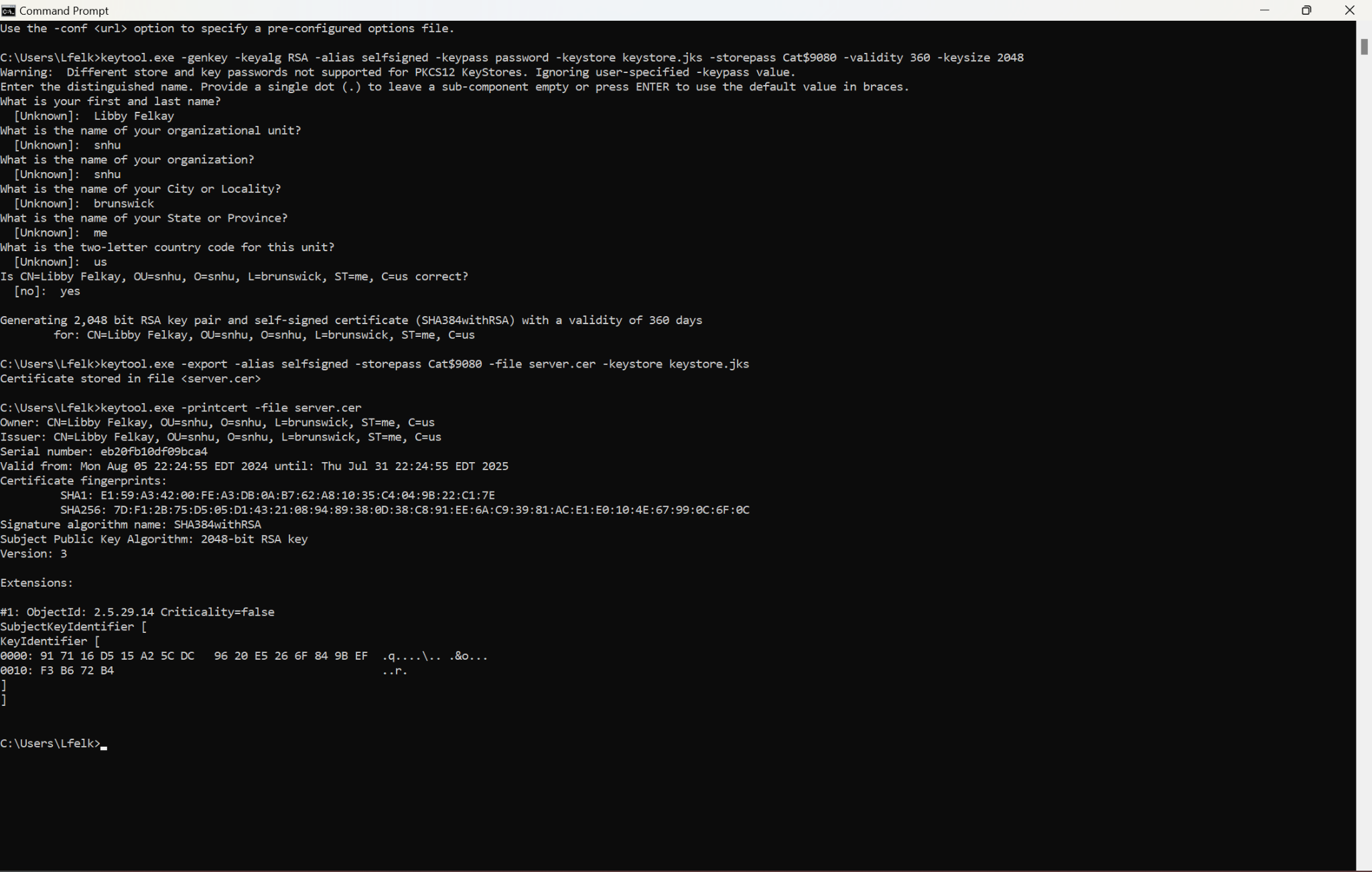
Random numbers are needed to make encryption unpredictable. Symmetric keys are used for both encrypting, and decrypting data. Non-symmetric keys are used with one public key and one private key. SHA as used here is non symmetric. AES is symmetric where it is the encryption and key.

Describe the history and current state of encryption algorithms.

The history of encryption goes back centuries, we used to make human codes to prevent enemy spies but now we have algorithms to conceal data for us. With the advancement of computers, we now have more capability to crunch numbers for more and more secure encryption. We currently use AES as the standard.

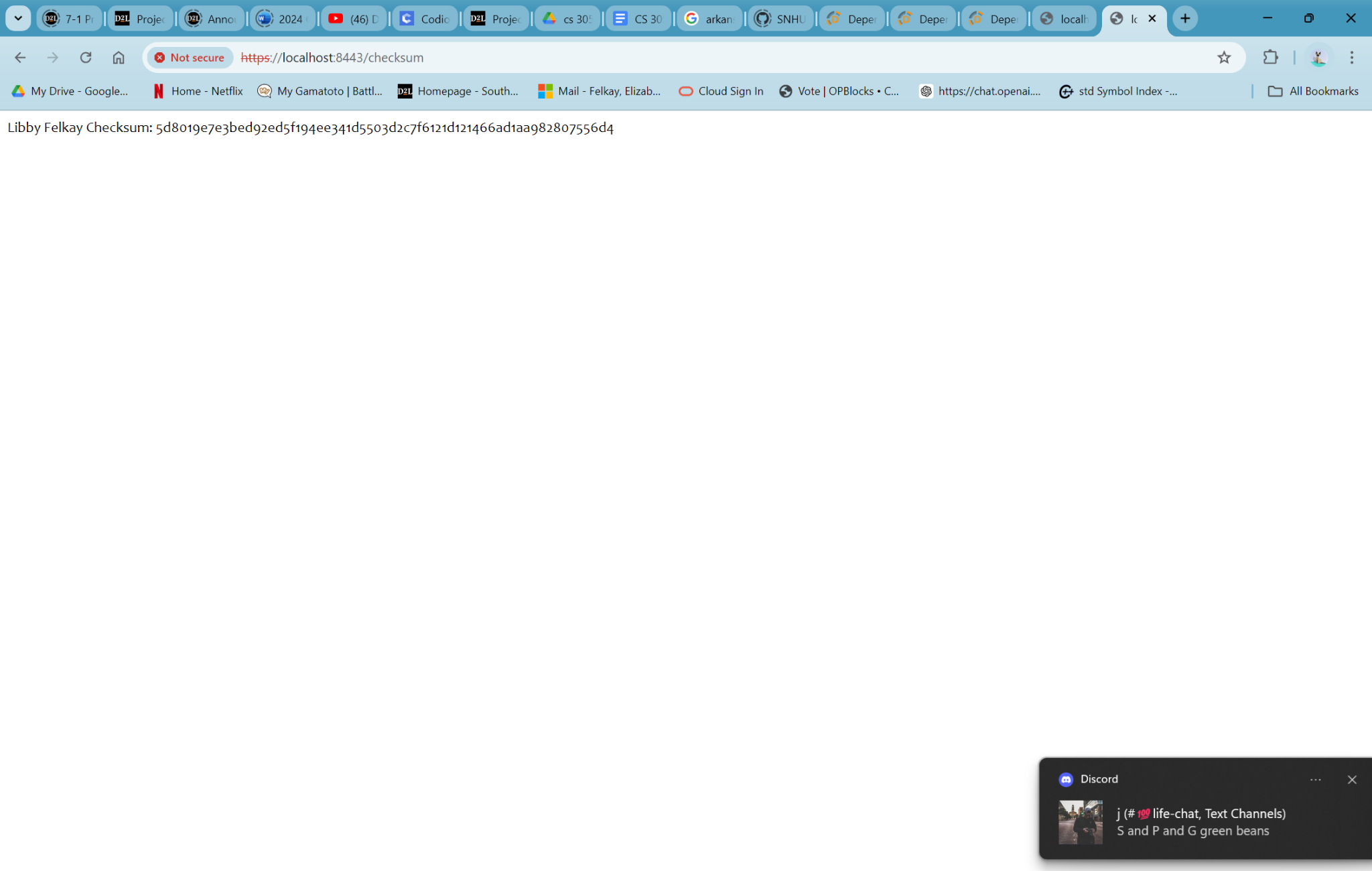
## Certificate Generation

Insert a screenshot below of the CER file.



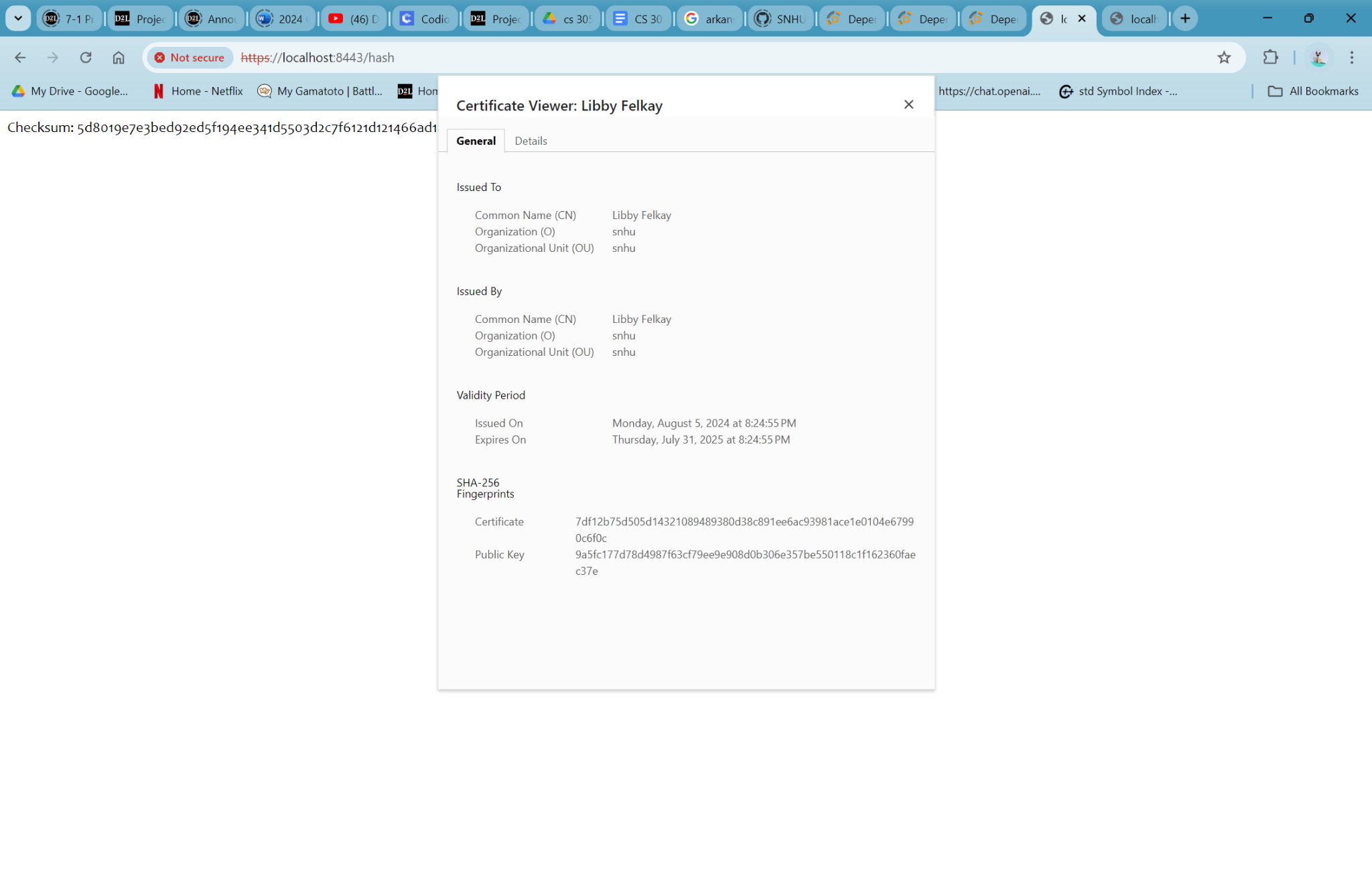
## Deploy Cipher

Insert a screenshot below of the checksum verification.



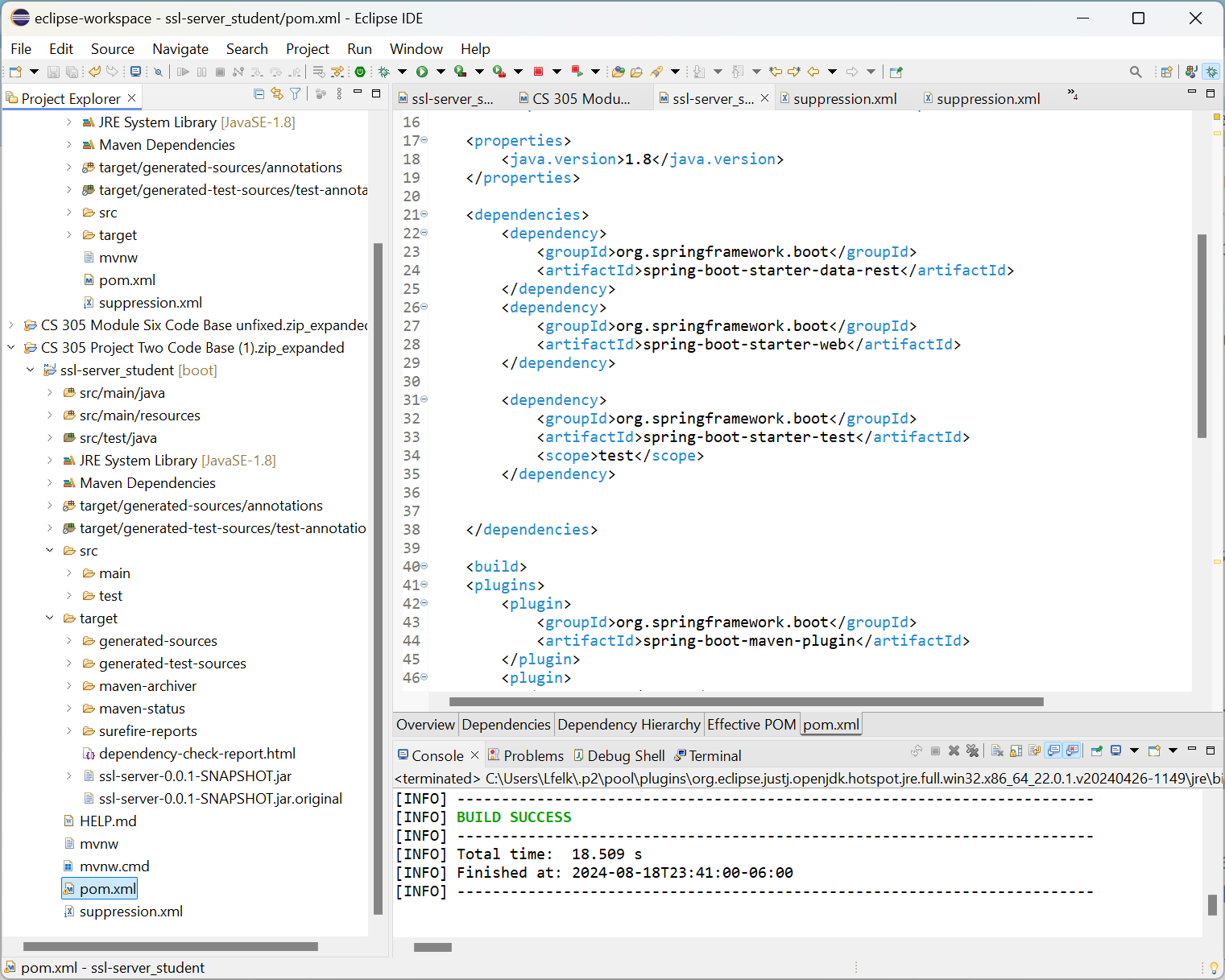
## Secure Communications

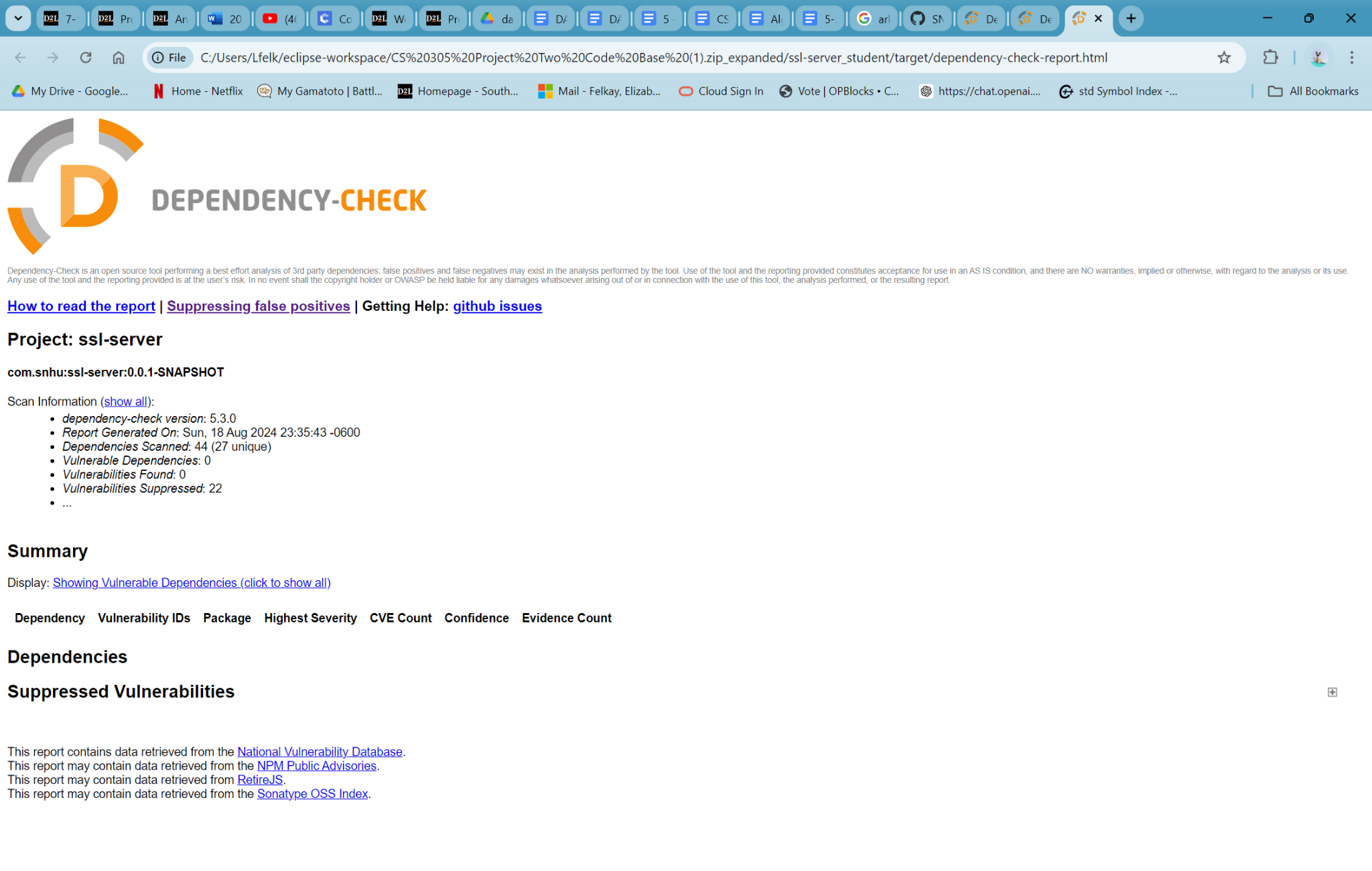
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

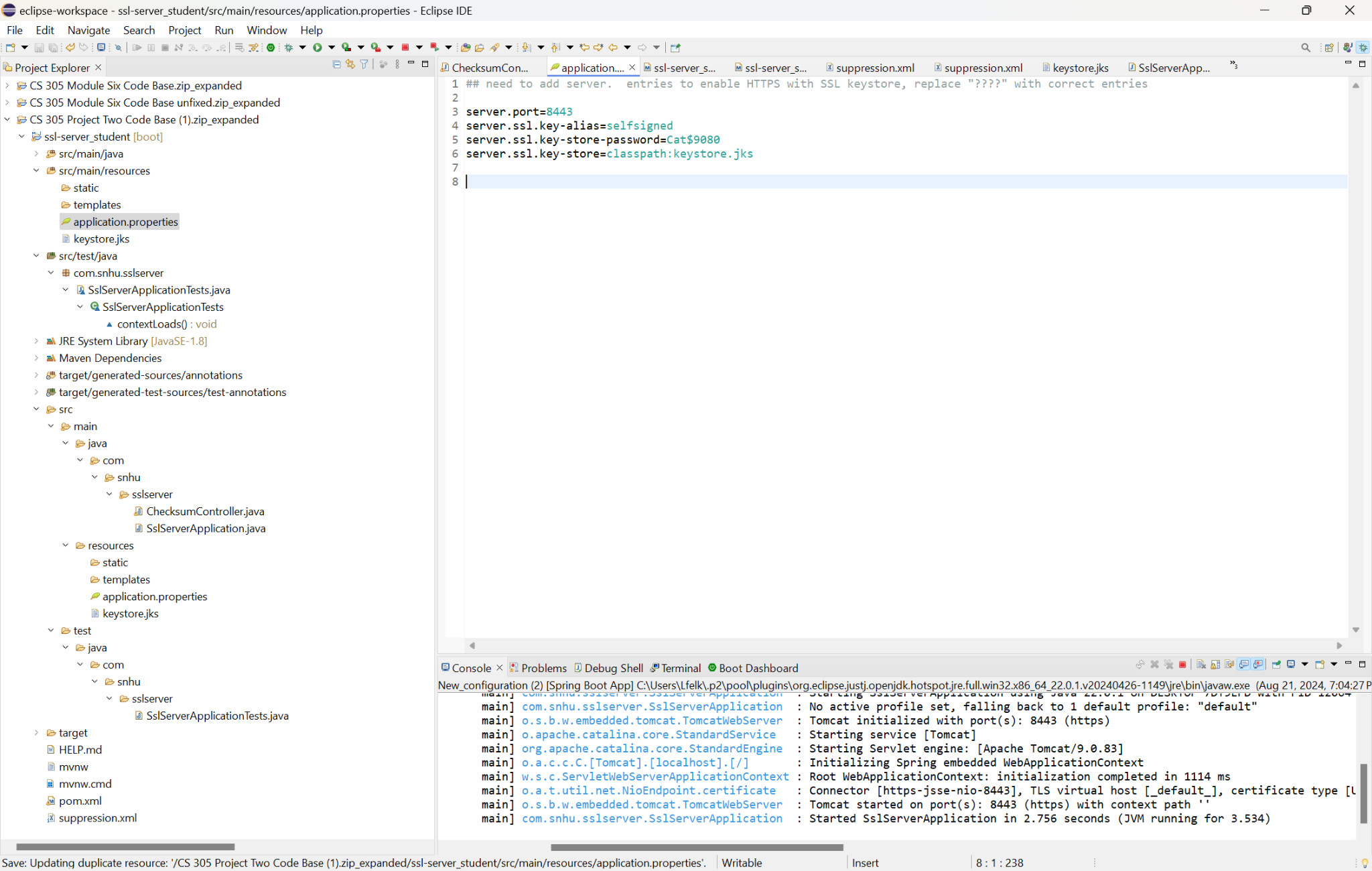
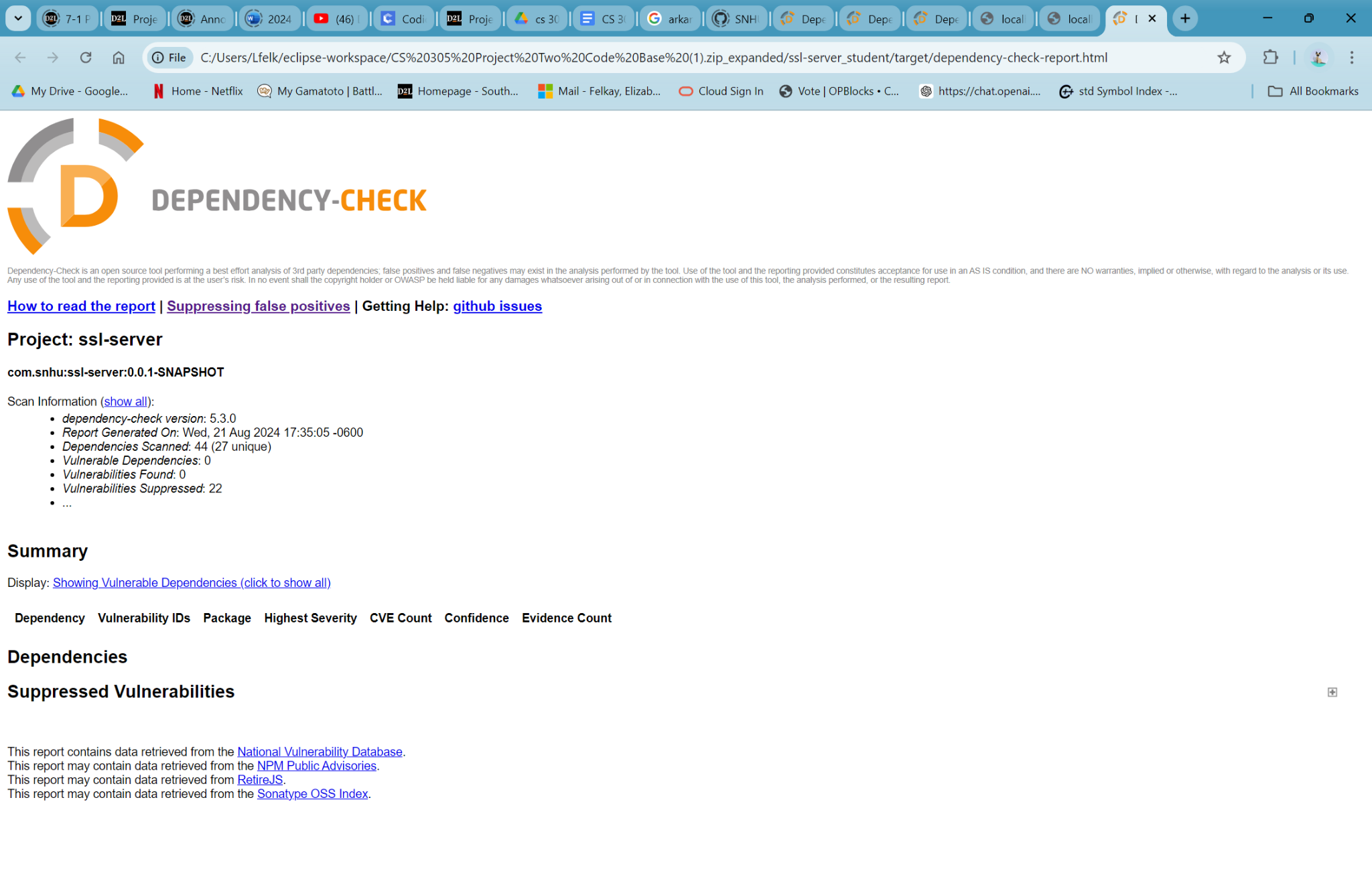
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

Insert a screenshot below of the refactored code executed without errors.



## Summary

Areas I addressed based on the vulnerability flowchart are using cryptography, authentication, APIs, and code error. I also updated out of date plugins such as springboot to resolve vulnerabilities. I used a checksum function to encrypt data using SHA-256(secure hashing algorithm) which is the best option for this program. I added a self signed key so a password is needed for authentication. There is an error catch for the catchsum so the program doesn't behave unexpectedly when unexpected data is entered. OWASP was used to detect known vulnerabilities and remedy the applicable ones that I could fix. The program runs https instead of http with the s meaning secure(google chrome doesn't like self signed so it shows as not secure)

## Industry Standard Best Practices

I followed industry best practices by using a plugin like OWASP to check for known vulnerabilities as well as keeping everything updated. The data is now also encrypted so even if it is breached, it won't be decipherable. Plugins and dependencies will usually add security fixes with each update so it’s important to keep them up to date. Following the industry best practices will help you make and keep programs secure. When this is facing client-side there should be input validation added such as only allowing letters and spaces for names and only numbers and commas for funds.